The Reward Problem: Where does the most important signal in reinforcement learning come from?

Anony Mouse
fake@email.com
Location
University

Abstract

Most reinforcement learning (RL) research is conducted with the presumption that there is a meaningful reward signal that is available to an agent at every moment in time, in any environment that the agent might find itself in. This reward signal is most often generated by a human explicitly programming a function that maps the agent's state observations and actions to rewards. If defining a reward function is a precondition to an agent's interaction in many different new environments, then RL in the real-world will not work. How might we scale up the diversity and complexity of intelligent and meaningful agent behaviours if each new goal or task requires significant human effort to define reward signals? Specifically, reward signals that, when leveraged as an optimization target for an RL algorithm, causes some behaviour towards some goals or purposes that human observers perceive as intelligent. This is The Reward Problem, it is unique and distinct from the RL Problem (i.e. given a reward signal, how do you optimize for it in a data efficient way?) The Reward Problem is unavoidable and critical to address immediately if we care at all about using RL algorithms in the real-world, especially for problems that we do not know how to solve ourselves.

1 Heeding The Bitter Lesson

Consider The Bitter Lesson (Sutton, 2019): "we have to learn the bitter lesson that building in how we think we think does not work in the long run." The implication here is that we should do everything in our power to remove ourselves from the solution space, as we are the limiting factor. This perspective is not new, in fact it was one of the earliest theories in the entire field of computing science. In 1945, Turing referred to it as 'the human brake', and said: "once the human brake is removed the increase in speed is enormous" (Hicks, 2008). Turing saw the incredible benefits of the future of computing. The promise of artificial intelligence is that much more complicated processes can be performed than could be handled by humans alone. And yet, in much RL research, we find ourselves at an impasse. We have to heed the warning of the bitter lesson, and remove the human brake. We have to stop designing and implementing reward functions by hand, and we have to come up with creative solutions to do so. We must apply the bitter lesson to reward design.

1.1 The success of reward design

This being said, reward design backstops many RL success stories. For example, Silver et al. (2016) discuss in two brief sentences the reward function that they used to master the game of Go with deep neural networks and tree search. In short, it is zero for all time steps, +1 for winning, and -1 for losing. In a sense, this simple reward function, when combined with deep RL and search, was all that was needed. But, the complexity of the reward function is hidden in the details; the winning and losing terminal state of Go is not actually clear and agreed upon state, and some decision on an appropriate approximation was required.[1]

As another example of the success of reward design, consider OpenAI's recent 5v5 Dota success (OpenAI Five). In the accompanying research paper (Berner et al., 2019), the authors describe how they tried to "assign reward for behaviours that [they] thought lead to the goal of winning the game, without over crafting the reward to [their] own expectations". The reward that they designed and used was a linear combination of multiple signals from the game, weighted in proportion to their subjective opinion of importance balanced with empirical results. The authors assured readers that the signals and weights were designed by Dota experts at the start of the research project, and they "have only been tweaked a handful of times since." They also share their experience training an agent with a simpler reward function. They ran an experiment where the agent was only rewarded for winning and penalized for losing: "it trained an order of magnitude slower, and somewhat plateaued in the middle, in contrast to the smooth learning curves with shaped rewards."

As a final example, consider the recent success from Sony AI in outracing champion Gran Turismo drivers with Deep RL. As they say in the paper (Wurman et al., 2022), "we construct a reward function that enables the agent to be competitive while adhering to racing's important, but under-specified, sportsmanship rules". As they say: "the agent is given progress reward for the speed with which it advanced around the track and penalties if it went out of bounds, hit a wall or lost traction." These are human communicable preferences, transformed into code by a designer and developer, and iterated on, because as they found, "shaping rewards allowed the agent to quickly receive positive feedback for staying on the track and driving fast." To their credit, they discuss in detail how observing resulting agent behaviour led them to modify the reward function over the course of their research project: "Progress reward alone is not enough to incentivize the agent to win the race. If the opponent was fast enough, the agent would learn to follow it and accumulate large rewards without risking potentially catastrophic collisions. Adding rewards specifically for passing helped the agent learn to overtake other cars." The take home message from each of these examples is reiterated in the paper from Sony (emphasis my own): "we spent a lot of time trying to figure out what things we needed to change in order to get that superhuman performance".

2 Why are things different now?

We have been working with human programmed reward functions, because we have been using programmed environments that we humans have designed and built. But, we do not program the world. So, asking the question "Where does the reward come from?" is not disingenuous. It points to an underlying concern about 'how the world is' when paired with the fact that we desire agents that do things in the real world. So, it is a question of curiosity: how might we program suitable rewards? Is it doable at all? If so, where will these rewards come from?

The question of where the reward comes from fundamental to RL research as most RL research focuses on the implications of reward on behaviour. It is a question posed without malice or intent to distract or offend. RL research ought to concern itself with questions about the source of numerical reward signals, and questions pertaining to the maximisation of those signals.

One might also ask: Where does the question "Where does the reward come from?" itself come from? It is not a new question posed in this paper, rather it is one that has been developing in parallel alongside progress in RL research for many years. In a 2007 paper, Doya mentioned that even then "the designing process takes a lot of trial and error" (Doya, 2007). The question was stated plainly as the title of a 2009 paper which explored how reward functions emerge and whether they ought to be considered as intrinsic or extrinsic to the agent (Singh et al., 2009).

[1] See the Tromp-Taylor rule set (https://senseis.xmp.net/?TrompTaylorRules) used for computer Go, which gives a computable score for any game in a well-defined way and which was used to train AlphaGo and successors. Most rule sets agree in most cases but there are some rare edge cases.

3 What is reward and where does it come from?

The Sutton and Barto textbook on Reinforcement Learning (RL) states plainly that: “Reinforcement learning is learning what to do so as to maximise a numerical reward signal” (pg. 1, Sutton & Barto (2018)). This reward signal is the definition of the RL problem, and it is the ‘primary basis for altering a policy’; the policy is the mapping from the state that agent is currently in to the next action the agent will take. The book goes on to describe how the rewards are basically given directly by the environment. This implicates the environment designer in the design of the reward. Where else is the reward to come from?

Previous research has advanced the position that the reward need not come from the environment. The environment is what is external to the agent, and all the environment does is proceed from state to next state, sometimes in response to the actions of the agent (if they are affectual) and sometimes without any regard for the agent(s) within. In this paradigm, the agent evaluates the reward within itself, in response to the changing environment (Oudeyer et al., 2007; Singh et al., 2009). The agent has no ability to directly modify the reward function within its own lifetime, rather, the function is constructed within an outer loop of evolutionary optimization.

One might ask at this juncture if we ought to prefer reward that is external from the agent and comes from the environment, or alternatively, if agent derived reward is preferable? In such a circumstance, it can help to consider the benefits and detriments of either alternative. If reward is external to the agent, and is functionally a component of the environment, this allows for fair comparison of the performance of multiple agents against the single ground-truth definition of the external reward. If, on the other hand, reward is part of the agent, this comparison is no longer possible. But, the benefit is that environment design need not include reward design, and, with any luck, the reward signal will be easier for the agent to interpret.

Does it matter where the reward comes from? I believe so, as does the RL textbook. In fact, Sutton & Barto (2018) state clearly that: “the success an RL application strongly depends on how well the reward signal frames the goal of the application’s designer and how well the signal assesses progress in reaching that goal.” It goes on, “for these reasons designing a reward signal is a critical part of any application of reinforcement learning”. So, it is clear that the message up until now is that the reward signal is a design component of the environment, and that it is of paramount importance to the success of agents trained with RL. This has led many RL researchers to ask the same question: “how do we design the part of the agent’s environment that is responsible for computing each scalar reward and sending it to the agent at each timestep?” (pg. 469, Sutton & Barto (2018)).

As the case studies in the previous section emphasise, just designing any reward function in an environment is one thing. But, designing a reward signal so that as an agent learns its behaviour approaches what the application designer actually wants is a much more challenging task. Not only challenging, but one that is of great import if RL is to be used in real-world applications given concerns about value alignment, and reward (mis)design and the undesirable and dangerous behaviours learned through reward hacking and specification gaming (Knox et al., 2022; Brown et al., 2021; Krakovna et al., 2020; Christian, 2021).

3.1 Rewards come from humans

Where does reward come from? Humans. But, why? Why do rewards come from humans? This question can be answered two ways. The first is by answering the question: Why do we care to build agents which optimise human designed reward functions at all? The answer is alignment: we’d like agents that behave in accordance with human preferences. The second way to answer the question is a little more philosophical: Why don’t rewards come from somewhere other than humans? This is addressed in more depth a little later on, but briefly, the answer is largely for historical reasons and convenience. Humans are the best source of reward design we have for now.

It might be helpful to consider a historical perspective to gain an appreciation of where the rewards have come from in previous RL research. The question of where the reward comes from is valid in the recent and complex environments of Dota and Go described above, but it is also valid in more classic, simple domains such as Mountain Car and Cart Pole. In a Cart Pole environment, the agent must move a cart left and right to balance a pole vertically. The code for this reward is explicitly defined as part of the environment in both the original pole.c and in the more recent Cart Pole implementation in Gymnasium. The reward is similar in both, and boils down to the longer the pole remains upright, the better the agent performs.

So, in the Cart Pole environment, where has the reward come from? The reward has come from humans. Some human(s) designed the dynamics of the environment (Barto et al., 1983). They decided what was good and what was better, and what was bad. They designed a reward signal around those decisions. They Implemented a function that computed that reward signal. That function provided reward to the agent at every timestep.

Assuming that 'the human(s) involved' answers the 'where' question, we can also ask 'how' questions. How did the human(s) convert their subjective opinion of quality into a reward signal interpretable by the agent? In the Cart Pole example, humans designed it by coding a computable reward signal for a specific goal in a specific environment. The same method of coding a computable reward function is what is used in many other environments in Gymnasium (Towers et al., 2023), and bsuite (Osband et al., 2020), and Chess, Go, etc. The prevailing trend in RL research is for the humans involved to decide what 'good behaviour' looks like and then program a reward function that an agent modulates its policy to maximise. This is considered a success when the resulting behaviour is deemed meaningful or intelligent by the humans involved.

This rather brutish coding method is not the only answer to the 'how' question. It is one way to construct an agent-facing proxy of human preference. It is not the only method by which a human expresses their subjective assessment of intelligent behaviour. It is useful when it is easy to codify a human preference of good behaviour. Inverse RL is another answer to the 'how' question. In inverse RL, the goal of the agent is to find a reward function from a set of expert demonstrations that could result in expert behaviour (Ng et al., 2000; Abbeel & Ng, 2004). That is, if an agent is provided with optimal trajectories which include only states and actions (i.e. no rewards) through some environment, can that agent recover a reward function that would cause such expert behaviours? This reward function could then be used to find a good policy – as in apprenticeship learning. Or, alternatively, some approximation of the expert policy could be learned directly using a supervised learning technique such as behavioural cloning (Pomerleau, 1988; Sammut et al., 1992). In this way, the agent could learn an implicit reward function based on the expert trajectories, without ever explicitly modelling the function itself.

Other answers to the 'how' question include Advice and Teaching (Chernova & Thomaz, 2014), and methods for interactively shaping an agent's behaviour based directly on human feedback, as in TAMER (Knox & Stone, 2009). These methods are useful when it is easy and relatively low cost to elicit human preferences. But, as humans are sometimes noisy, lazy, and/or inconsistent, often these methods build a human reward model to smooth out the human-delivered reward signal. There are many more methods for learning reward functions (Sumers et al., 2023), including: 1) learning from observed actions and inferring rewards, 2) learning from feedback such as comparisons, sketches, or clickers (as in TAMER), 3) learning from language-based instructions (e.g. converting human instructions into accomplishable goals), 4) learning from descriptions in existing bodies of text (e.g. reading the instructions for a video game to understand how to get a high score), 5) learning through pedagogy as in conversational active teaching and querying in uncertain contexts, and 6) learning from a large set of expressed human preferences (Christiano et al., 2017).

So, the reward comes from humans, and there are many methods that we can convert our preferences over a set of agent behaviours. And, many of the resulting reward functions would ultimately lead to the same, or similar, behaviour. Recently, there has been some work developing principles of reward design, preferring reward functions that are quickly learnable, or more-easily human interpretable (Sowerby et al., 2022).

In addition to questions of 'where' and 'how', once we appreciate that rewards for agents come from humans, it is natural to ask: "Who do these rewards come from?" Historically, 'RL researchers and their design decisions have answered the' who' question. Rewards for agents have come from expert game designers (as in Chess, Go, Shogi, Stratego, and Diplomacy), or researchers in RL and in control theory. More recently, as in the Gran Turismo example above, the rewards come from domain experts. As in the Christiano et al. (2017) paper, the rewards might be learned from a large set of non-domain expert preferences. There are important socio-technical considerations and questions of impact and ethics when we consider who is designing and developing reward functions and who is impacted by the behaviour of agents which learn from these reward signals, especially if these groups of individuals do not overlap.

3.2 What rewards are there in the real world?

As we design and build general agents these agents will interact in the real world. And, we hope these agents will act intelligently and meaningfully in the real world. In RL research, we are moving from a set of environments (e.g. Dota, Cart Pole, Chess, Go, Gran Turismo) to one massive complex environment: the real world. Given that we do not have access to change the reward function for the real world, how ought we proceed? To answer this question, it is important to restate what we are ultimately interested in: agents that behave intelligently as they interact in the real world.

To make progress towards, we can start by optimising for what we are ultimately interested in. This implies that agents ought to interact in the real world sooner rather than later. It also implies that we ought to build mechanisms whereby agents have an opportunity to behave intelligently while humans observe them either during or post-interaction. These situations ought to be those where humans can express their opinion of more or less intelligent behaviour, but for which it is hard to encode such an opinion. If the opinion were easy to express as a reward-delivery function in such a way that an agent maximising the signal would exhibit intelligent behaviour, then such an alternative approach might be preferred.

4 Where might the reward come from in the future?

The rewards of the future might come from multiple sources. They might come from multiple humans in a collective community, or the reward functions might evolve due to an evolutionary process within the agent. Agents might collect trajectories in an environment without a pre-specified reward function, and then attempt to compute optimal policies for a collection of reward functions (Jin et al., 2020). Alternatively, the reward function might be based on an internal monologue within the agent itself, or other agents in the environment might propose reward functions for each other. These ideas may sound far fetched, but there is already research that explores non-human sources of reward signals (Schmidhuber, 2010; Singh et al., 2010; Niekum et al., 2011; 2010; Eysenbach et al., 2018; Wang et al., 2019).

Evolution may be a way forward, and it has led to what many consider to be intelligent behaviour in humans. So, can agents use search and learning to find reward functions which lead to meaningful behaviour? Would such agents be able to generate reward signals in one way or another without direct or indirect human interaction? I think such a reasonable approach could work, but it might take quite a long time. As well, natural selection has generated biological reward signals in accordance with some fitness functions (e.g. stay fit and have fun) – Where do these come from? Primary motivators potentially (e.g. get food and find friends) – But, this begs the question: What are the constraints and primary motivators on these artificial agents? Where do they come from?

However, considering these angles opens up interesting research questions. For example, what might other reward functions look like for the Atari Learning Environment? Could alternative reward functions lead to faster learning or better transfer between games? Then again, in the end, who is to say that the resulting behaviours would be intelligent, meaningful, or interesting? Humans would be the ultimate subjective assessor of such qualities.

I believe that rewards will continue to come from humans, as they always have. Reward is defined by human subjective assessment, just as it has been in every RL environment that I know of – please correct me if you know otherwise. Thus, we want better ways to convert subjective assessment (e.g. human preferences) into reward signals that RL agents can optimise for. One promising approach is to build massive datasets and learn models of reward signals from that data. If we have these datasets, is RL even necessary? Couldn't we just do imitation learning? Imitation learning might tell us what actions to take in similar situations. But, learning the underlying rewards might give us more information about something more fundamental.

5 How do we get subjective preference data from humans?

Ask them. To elicit preferences, we will need to design new and interesting methods to ask humans many different questions. For example, we can ask: 1) to define goal states that they want agents to reach, 2) to define trajectories through states that they want agents to traverse, 3) to definite tasks they wants agents to perform, 4) to observe trajectories, describe what was accomplished, and assess success, 5) to describe what they expect an intelligent agent might do, 6) to rank a set of trajectories. While many of these questions lead to illumination of a human's preference over behaviours, we might also use answers from humans to safely guide the efficient exploration of agents – RL works best when the agent can receive reward, that is base competencies lead to rewarding states. There are many more modes of interaction which could enable agents to learn how to behave from humans, and I believe that RL research in this area is absolutely fundamental to understanding intelligence.

Converting human preferences into reward models is a hard modelling problem. To understand why, consider what happens if people act irrationally or inconsistently – I'd argue this is a common occurrence. People often disagree with each other and sometimes disagree with themselves from one day to the next. While this does not preclude us from attempting to model human preferences, it makes doing so an important research challenge and one that should include mechanisms for continual learning, continual adaptation, and alignment with shifting preferences and values.

6 More Recent Examples

There are many recent examples of leveraging human preferences for training of RL models. In Fan et al. (2022), the authors noted that the cost of meticulously crafting dense rewards was too high. They note that agents developed in popular RL benchmarks often rely on these task-specific reward functions to guide random exploration. Therefore they used a combination of programmatic tasks, human brainstorming, and large language models (GPT-3, (Brown et al., 2020)), to construct tasks and data to learn a dense, language-conditioned, open-vocabulary, multi-task reward function from YouTube videos and their transcripts. This function computes a correlation between a language goal and a 16-frame video snippet. This correlation was then used as a reward function to train a strong multi-task RL agent.

Similarly, Ziegler et al. (2019) and Ouyang et al. (2022) both follow a similar pattern of using human preferences to optimise a policy with a reward model using reinforcement learning from human feedback or RLHF. Both of these works used a large pre-trained language model and a dataset of human preferences illustrating how RL can be a complimentary learning component to supervised learning. In these works, the reward is derived from human labelers who rank outputs from a model from best to worst. These rankings are used to train a reward model which can calculate a reward for an output to an unseen input. Rewards are used to update the underlying policy using the proximal policy optimization RL algorithm. There are many more works extending RLHF research in various ways for adapting language models to particular use cases (Nakano et al., 2022; Wu et al., 2021; Menick et al., 2022). As we continue to see performance improvements from RL fine-tuning of language models, there are even advances to remove the human-preference elicitation step by using RL from AI feedback, or using models to rank the quality of a set of samples (Bai et al., 2022).

7 Countering Conterpoints

I think it is valuable to consider alternative viewpoints to those presented in this article. As such, I attempt to address several of the objections that have been raised in this section.

First, RL is focused on researching algorithms regardless of the reward function, any reward will do and we should not concern ourselves with where it comes from. To this, I often ask how one knows if an algorithm will work on any reward function if it is not tested on functions that might elicit interesting behaviours. As well, how can the universality of an algorithm be claimed, when the only evidence of success is in circumstances when the signal was not only known but designed and defined by the interrogator. This response is often met with the retort that it is better to study questions in RL in isolated settings, with easy-to-code and understand reward functions first. To which I'd argue that toy problems are misleading and actively delay and distract from addressing the hard problem of satisfying human preferences by behaving interestingly and meaningfully in the real-world.. I further stress that this mindset incentives working on solving classes of reward functions that are very unlikely to appear in the real-world.

Others have said that asking questions like "where does the reward come from?" is like asking "where do the labels in supervised learning come from?" Why don't we ask similar questions to researchers in supervised learning? I'd argue that labels most often have come from humans, but automated labelling exists and is becoming more common. As well, I'd argue that with Model and Dataset cards, these questions are more well interrogated in Supervised Learning than in RL research. When I point this out, I've heard the response that perhaps designing reward functions is 'not our problem' – to this, I'd say: Whose problem is it? We are all involved in the interaction with the machine learning systems that we design, develop, and disseminate, should we not feel responsible for the reward function that results in the behaviour exhibited by the agents we train? The follow up that I have heard at this point is something like: "will the reward always come from humans?" to which I reiterate: perhaps, but not necessarily. We could imagine agents that reward each other, or even design each others' reward functions, but it is probably important what humans think while humans are around.

What about when humans were not around? Before humans existed, there was likely some form of reinforcement learning. And it is likely that RL will exist after humans do. In such cases, do human subjective preferences really matter? Likely not, in the same way that if the only living organism on earth was a single tree, and it fell down, the sound made would not matter. But, starting about 20,000 years ago, what humans thought started to be relevant for other intelligent beings near earth, and that is going to continue for the foreseeable future. What about intelligent systems far from earth? They might be doing RL well out of range of our pesky subjective preferences. But, if we ever come into contact with them, if we ever share an environment, we might impose our assessments of intelligence and meaningfulness on their behaviours in the real-world.

8 Concluding Thoughts

There are two questions that often get conflated in RL research:

- What should an agent do to maximise a given reward signal?
- How do we translate our soft specifications for intelligent behaviours into reward signals for agent maximisation?

The first question is "The RL Problem": given a reward signal, how do you optimise for it in a data efficient way? This question views RL as a common model of an intelligent agent, and it assumes that we agree on the problem to solve. That is, we have a reward signal. It assumes that the solution to the problem can be learned by maximising a given reward signal, and RL can be used to maximise the signal.

The second question is “The Reward Problem”. It is the question of where the reward comes from, how it is designed, who designs it, and why it might be useful to design in one manner versus another. The Reward Hypothesis speculates that all goals may be representable by a reward function. But, even if that is true it doesn’t mean that the reward function can be learned; finding a reward function which represents a goal may be intractable, and some ways of specifying goals may be preferable (Roy et al., 2021). The Reward Hypothesis also doesn’t imply that the reward function can be learned from; a reward function may be so sparse that a rewarding state is never reached for all eternity – RL doesn’t work as well when the reward is difficult to acquire for the agent, or when the reward is difficult to codify or curate data to support on the human-side. Finally, intelligent and meaningful behaviour can depend (as much of it does) on human interaction, or “we’ll know it when we see it”. The only evaluation you can not get rid of is interaction with humans.

I’d argue that we’ve spent a great deal of time and energy on the first question, and not nearly enough on the second question. The reward problem is unavoidable and fundamental if we care at all about using our RL algorithms.

Acknowledgements

Use unnumbered third level headings for the acknowledgements. All acknowledgements, including those to funding agencies, go at the end of the paper. Only add this information once your submission is accepted and deanonymized.

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Yang Bai, Desen Zhou, Songyang Zhang, Jian Wang, Errui Ding, Yu Guan, Yang Long, and Jingdong Wang. Action quality assessment with temporal parsing transformer, 2022.

Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5):834–846, 1983.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Daniel S. Brown, Jordan Schneider, Anca D. Dragan, and Scott Niekum. Value alignment verification, 2021.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Sonia Chernova and Andrea L Thomaz. *Robot learning from human teachers.* Morgan & Claypool Publishers, 2014.

Brian Christian. *The alignment problem: How can machines learn human values?* Atlantic Books, 2021.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Kenji Doya. Reinforcement learning: Computational theory and biological mechanisms. *HFSP journal*, 1(1):30, 2007.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function, 2018.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35: 18343–18362, 2022.

Mar Hicks. Repurposing turing's" human brake". *IEEE Annals of the History of Computing*, 30(4): 108–108, 2008.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pp. 4870–4879. PMLR, 2020.

W Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pp. 9–16, 2009.

W. Bradley Knox, Alessandro Allievi, Holger Banzhaf, Felix Schmitt, and Peter Stone. Reward (mis)design for autonomous driving, 2022.

Victoria Krakovna, Jonathan Uesato, Vladimir Mikulik, Matthew Rahtz, Tom Everitt, Ramana Kumar, Zac Kenton, Jan Leike, and Shane Legg. Specification gaming: the flip side of AI ingenuity — deepmindsafetyresearch.medium.com. https://deepmindsafetyresearch.medium.com/specification-gaming-the-flip-side-of-ai-ingenuity-c85bdb0deeb4, 2020. [Accessed 29-04-2024].

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, and Nat McAleese. Teaching language models to support answers with verified quotes, 2022.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022.

Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Scott Niekum, Andrew G Barto, and Lee Spector. Genetic programming for reward function search. *IEEE Transactions on Autonomous Mental Development*, 2(2):83–90, 2010.

Scott Niekum, Lee Spector, and Andrew Barto. Evolution of reward functions for reinforcement learning. In *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, pp. 177–178, 2011.

Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepesvari, Satinder Singh, Benjamin Van Roy, Richard Sutton, David Silver, and Hado Van Hasselt. Behaviour suite for reinforcement learning, 2020.

Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2):265–286, 2007.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Julien Roy, Roger Girgis, Joshua Romoff, Pierre-Luc Bacon, and Christopher Pal. Direct behavior specification via constrained reinforcement learning. *arXiv preprint arXiv:2112.12228*, 2021.

Claude Sammut, Scott Hurst, Dana Kedzier, and Donald Michie. Learning to fly. In *Machine Learning Proceedings 1992*, pp. 385–393. Elsevier, 1992.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE transactions on autonomous mental development*, 2(3):230–247, 2010.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Satinder Singh, Richard L Lewis, and Andrew G Barto. Where do rewards come from. In *Proceedings of the annual conference of the cognitive science society*, pp. 2601–2606. Cognitive Science Society, 2009.

Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.

Henry Sowerby, Zhiyuan Zhou, and Michael L Littman. Designing rewards for fast learning. *arXiv preprint arXiv:2205.15400*, 2022.

Theodore R Sumers, Mark K Ho, Robert D Hawkins, and Thomas L Griffiths. Show or tell? exploring when (and why) teaching with language outperforms demonstration. *Cognition*, 232: 105326, 2023.

Richard Sutton. The bitter lesson. *Incomplete Ideas (blog)*, 13(1):38, 2019.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023. URL https://zenodo.org/record/8127025.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions. *arXiv preprint arXiv:1901.01753*, 2019.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback, 2021.

Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896): 223–228, 2022.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.